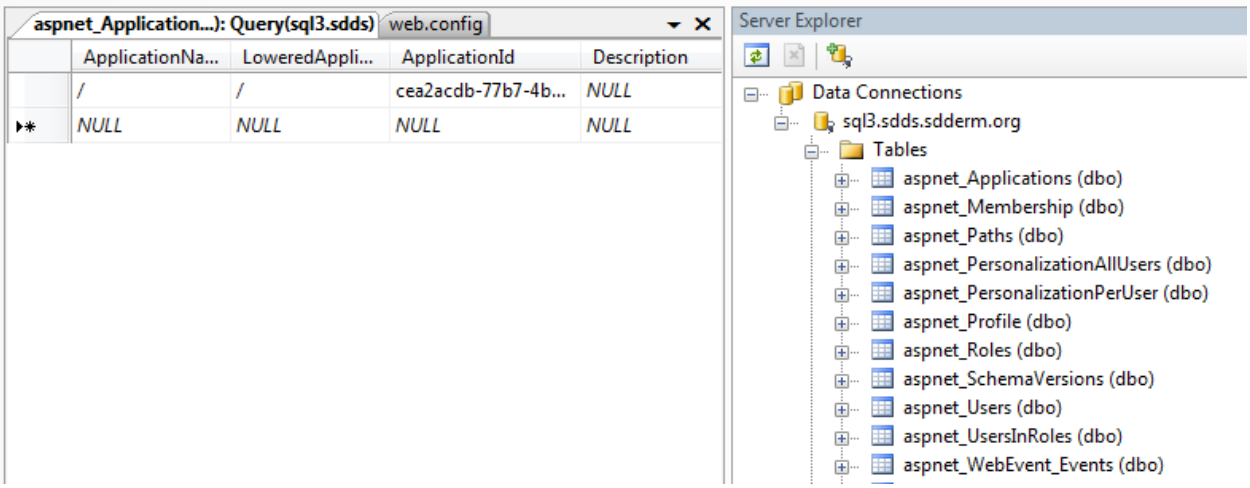
Validation of Viewstate MAC failed

The Validation of Viewstate MAC failed error commonly occurs when an ASP.NET application developer uses the Membership, Roles and/or Profile Providers included with the .NET Framework. The usual scenario is the developer creates the application on their local machine and stores the project files in a local directory. When the developer copies or publishes the project files to another local directory or to a remote server, the login system doesn't appear to function properly.

The solution is to manually add the applicationName attribute to the <providers> node of your web.config file and give it a value. By default ASP.NET auto generates the applicationName using your project's root directory path if the applicationName attribute isn't declared in your web.config. The applicationName is important because your ASP.NET Application Service database uses it in a number of tables. The applicationName is a relatively long string of characters and will look similar to: cea2acdb-97b7-4b58-ad12-22efg3bd582q.

If you move your project to any other directory either local or remote, a new applicationName string will be auto generated and added to your ASP.NET Application Service database. Any new users, roles, or profiles you create will be tied to the new applicationName. Once you move your application to a new directory, any users, roles, or profile information you created while your project resided in your previous directory will not function or throw the following error: "Validation of viewstate MAC failed. If this application is hosted by a Web Farm or cluster, ensure that <machineKey> configuration specifies the same validationKey and validation algorithm. Auto Generate cannot be used in a cluster." Each time you move your project to a different directory you'll run into this problem.

To solve this, first go to the Server Explorer in Visual Studio and open the database containing your ASP.NET Application Service database. Right click on the table named aspnet_Applications (dbo) and select "Show Table Data". Look to see what your ApplicationName and LoweredApplicationName fields are named. It should be "/". If it's named anything other than "/", rename both fields "/" as shown below.



Next open your web.config file. Add the applicationName attribute to your provider declaration(s). You should set the applicationName attribute equal to "/" to match the "/" fields in your aspnet_Applications (dbo) table. An example is provided below:

```
<roleManager enabled="true" defaultProvider="CustomizedRoleProvider">
      <providers>
        <add name="CustomizedRoleProvider"
             type="System.Web.Security.SqlRoleProvider"
             connectionStringName="MyConnectionString"
             applicationName="/"
        />
      </providers>
    </roleManager>

    <membership defaultProvider="CustomizedMembershipProvider">
      <providers>
        <add name="CustomizedMembershipProvider"
             type="System.Web.Security.SqlMembershipProvider"
             connectionStringName="MyConnectionString"
             applicationName="/"
             minRequiredPasswordLength="5"
             minRequiredNonalphanumericCharacters="0"
        />
      </providers>
    </membership>
```

Make sure you add the applicationName attribute to all your users, roles, profile, and/or any other <providers> nodes in your web.config or you'll run into problems. In the example above I'm using both the Role and Membership providers so I added it to both declarations.

Once you declare the applicationName attribute, ASP.NET will always use that application name when writing and connecting to the ASP.NET application service database regardless of where your application resides.

Steve Kozyk
CEO/Founder ITegrity
skozyk[at]itegritygroup.com
www.itegritygroup.com